```
   SSSSSSSSSSSS  UUU         UUU  MMM         MMM
   SSSSSSSSSSSS  UUU         UUU  MMM         MMM
   SSSSSSSSSSSS  UUU         UUU  MMM         MMM
SSS              UUU         UUU  MMMMMM   MMMMMM
SSS              UUU         UUU  MMMMMM   MMMMMM
SSS              UUU         UUU  MMMMMM   MMMMMM
SSS              UUU         UUU  MMM   MMM   MMM
SSS              UUU         UUU  MMM   MMM   MMM
SSS              UUU         UUU  MMM   MMM   MMM
   SSSSSSSSS     UUU         UUU  MMM         MMM
   SSSSSSSSS     UUU         UUU  MMM         MMM
   SSSSSSSSS     UUU         UUU  MMM         MMM
            SSS  UUU         UUU  MMM         MMM
            SSS  UUU         UUU  MMM         MMM
            SSS  UUU         UUU  MMM         MMM
            SSS  UUU         UUU  MMM         MMM
            SSS  UUU         UUU  MMM         MMM
            SSS  UUU         UUU  MMM         MMM
SSSSSSSSSSSSS    UUUUUUUUUUUUUUU  MMM         MMM
SSSSSSSSSSSSS    UUUUUUUUUUUUUUU  MMM         MMM
SSSSSSSSSSSSS    UUUUUUUUUUUUUUU  MMM         MMM
```

```
_S


Sy
--
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
LI
ME
ME
RM
RM
SU
SU
SU
SU
SU
SU
SU

SU
SU
SU
SU
SU
SU
SU
SU
SU
SU
SY
SY
SY
SY
SY
SY
```

```
**FILE**ID**SUMFILES                                    G  5
```

```
  SSSSSSSS  UU      UU  MM      MM  FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    SSSSSSSS
  SSSSSSSS  UU      UU  MM      MM  FFFFFFFFFF    IIIIII    LL          EEEEEEEEEE    SSSSSSSS
SS          UU      UU  MMMM  MMMM  FF              II      LL          EE          SS
SS          UU      UU  MMMM  MMMM  FF              II      LL          EE          SS
SS          UU      UU  MM  MM  MM  FF              II      LL          EE          SS
SS          UU      UU  MM  MM  MM  FF              II      LL          EE          SS
  SSSSSS    UU      UU  MM      MM  FFFFFFFF        II      LL          EEEEEEEE      SSSSSS
  SSSSSS    UU      UU  MM      MM  FFFFFFFF        II      LL          EEEEEEEE      SSSSSS
        SS  UU      UU  MM      MM  FF              II      LL          EE                  SS
        SS  UU      UU  MM      MM  FF              II      LL          EE                  SS
        SS  UU      UU  MM      MM  FF              II      LL          EE                  SS      ::::
        SS  UU      UU  MM      MM  FF              II      LL          EE                  SS      ::::
SSSSSSSS    UUUUUUUUUU  MM      MM  FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS        ::::
SSSSSSSS    UUUUUUUUUU  MM      MM  FF            IIIIII    LLLLLLLLLL  EEEEEEEEEE  SSSSSSSS        ::::

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

SUM
V04

```
          0000  ;
          0000  ; Version:        'V04-000'
          0000  ;
          0000  ;****************************************************************************
          0000  ;*                                                                          *
          0000  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
          0000  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
          0000  ;*  ALL RIGHTS RESERVED.                                                    *
          0000  ;*                                                                          *
          0000  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
          0000  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
          0000  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
          0000  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
          0000  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
          0000  ;*  TRANSFERRED.                                                            *
          0000  ;*                                                                          *
          0000  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
          0000  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
          0000  ;*  CORPORATION.                                                            *
          0000  ;*                                                                          *
          0000  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
          0000  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
          0000  ;*                                                                          *
          0000  ;*                                                                          *
          0000  ;****************************************************************************
          0000  ;
          0000  ;
          0000  ; Assembly parameters
          0000  ;
00000200  0000          BUF_SIZE = 512                  ; Size in bytes of slipr input buffers
00000084  0000          CMD_SIZE = 132                  ; Size of input command line
          0000  ;
          0000          $NAMDEF
          0000          $RABDEF
          0000          $FABDEF
          0000          $CLIDEF
          0000  ;
          0000  ; Edit node offsets
          0000  ;
00000000  0000          ED$L_FWD    = 0                 ; Forward pointer
00000004  0000          ED$L_BWD    = 4                 ; Backword pointer
00000008  0000          ED$W_LOC1   = 8                 ; Locator 1
0000000A  0000          ED$W_LOC2   = 10                ; Locator 2
0000000C  0000          ED$W_LINES = 12                 ; Insert lines
0000000E  0000          ED$W_RFA    = 14                ; Record file address (3 words)
00000014  0000          ED$L_FILE   = 20                ; File node pointer
00000018  0000          ED$B_FLAGS = 24                 ; Flags
00000019  0000          ED$B_FILENO= 25                 ; File number
          0000  ;
0000001A  0000          ED$K_BLN    = 26
          0000  ;
          0000  ;
          0000  ; File node offsets
          0000  ;
00000000  0000          SLP$L_FWD  = 0                  ; Forward pointer
00000004  0000          SLP$L_BWD  = 4                  ; Backward pointer
00000008  0000          SLP$W_LOC1 = 8                  ; Locator-1
```

```
0000000A  0000          SLP$W_LOC2 = 10                     ; Locator-2
0000000C  0000          SLP$B_FLAGS= 12                     ; Flags
0000000D  0000          SLP$B_FILENO = 13                   ; File priority
0000000E  0000          SLP$W_DOT  = 14                     ; Dot value
00000010  0000          SLP$Q_AUDDS= 16                     ; Audit string desciptor
00000018  0000          SLP$T_AUDST= 24                     ; Audit string
00000028  0000          SLP$Q_AUCDS= 40                     ; Current audit string descriptor
00000030  0000          SLP$T_AUCST= 48                     ; Current audit string
00000040  0000          SLP$Q_CMNT = 64                     ; Comment descriptor
00000048  0000          SLP$T_NAM  = 72                     ; NAM block
          0000    ;
000000A8  0000          SLP$K_BLN  = SLP$T_NAM + NAM$K_BLN
          0000    ;
          0000    ;
          0000    ; Macro to print error message
          0000    ;
          0000          .MACRO  ERRMSG  NAME,LIST
          0000          $$ = 0
          0000          .IRP    L,<LIST>
          0000          PUSHL   L
          0000          $$=$$+1
          0000          .ENDR
          0000          PUSHL   #$$
          0000          MOVL    #MER$_'NAME',R0
          0000          PUSHL   R0
          0000          CALLS   #$$+2,G^LIB$SIGNAL
          0000          .ENDM   ERRMSG
```

```
                           0000          .TITLE  SUMFILES
                           0000          .IDENT  /V04-000/
                           0000
                           0000  ;
                           0000  ;****************************************************************************
                           0000  ;*                                                                          *
                           0000  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
                           0000  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
                           0000  ;*  ALL RIGHTS RESERVED.                                                    *
                           0000  ;*                                                                          *
                           0000  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
                           0000  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
                           0000  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
                           0000  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
                           0000  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
                           0000  ;*  TRANSFERRED.                                                            *
                           0000  ;*                                                                          *
                           0000  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
                           0000  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
                           0000  ;*  CORPORATION.                                                            *
                           0000  ;*                                                                          *
                           0000  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
                           0000  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
                           0000  ;*                                                                          *
                           0000  ;*                                                                          *
                           0000  ;****************************************************************************
                           0000  ;
                           0000  ; Procedure to prompt user to supply a list of input files
                           0000  ; and a single output file.  At least one input file must be
                           0000  ; supplied.  The procedure will continue to prompt for input files
                           0000  ; until at least one is supplied.  The single output file
                           0000  ; is optional
                           0000
                           0000          $NAMDEF
                           0000          $FABDEF
                           0000  ;
                           0000  ;
                           0000  ;
                       00000000          .PSECT  $CODE,EXE,NOWRT
                           0000
                           0000  GET_FILES::
                      0000 0000          .WORD   0
      6D    0000'CF    DE  0002          MOVAL   W^GET_HANDLER,(FP)       ; Set condition handler
                           0007  10$:
 0030'CF    0001'CF    DE  0007          MOVAL   W^PROMPT_INPUT+1, -      ; Set up read prompt string
                           000E                  W^CMD_INPUT_RAB+RAB$L_PBF
 0034'CF    0000'CF    90  000E          MOVB    W^PROMPT_INPUT, -
                           0015                  W^CMD_INPUT_RAB+RAB$B_PSZ
                 1C    10  0015          BSB     INPUT_FILES              ; Get input files
              18 50    E9  0017          BLBC    R0,20$                   ; If any errors start again
                 5B    D5  001A          TSTL    R11                      ; If zero input files given reprompt
                 E9    13  001C          BEQL    10$
 0030'CF    0001'CF    DE  001E          MOVAL   W^PROMPT_OUTPUT+1, -     ; Set up 'Output' prompt string
                           0025                  W^CMD_INPUT_RAB+RAB$L_PBF
 0034'CF    0000'CF    90  0025          MOVB    W^PROMPT_OUTPUT, -
                           002C                  W^CMD_INPUT_RAB+RAB$B_PSZ
               0167    30  002C          BSBW    OUTPUT_FILE              ; Get output file
```

```
                  00 50   E9  002F     58           BLBC    R0,20$                 ; If any errors start again
                              0032     59 20$:
                          04  0032     60           RET
```

```
                              0033    62          .SBTTL  INPUT_FILES
                              0033    63  ;
                              0033    64  ;
                              0033    65  ; Subroutine to get input files
                              0033    66  ;
                              0033    67  ; Inputs:
                              0033    68  ;         None
                              0033    69  ;
                              0033    70  ; Outputs:
                              0033    71  ;         R0 = Success/error status
                              0033    72  ;
                              0033    73  ;
                              0033    74  INPUT_FILES:
                 5B     D4    0033    75          CLRL    R11                             ; Initialise input files count
  0030'CF   0004'CF     D0    0035    76          MOVL    W^DEF_NAME+4,W^INPUT_FAB+FAB$L_DNA ; Set default file name
  0035'CF   0000'CF     90    003C    77          MOVB    W^DEF_NAME,W^INPUT_FAB+FAB$B_DNS
                              0043    78  10$:
       56   0000'CF     DE    0043    79          MOVAL   W^INPUT_BUF,R6                  ; Set address to put file name string
               01AB     30    0048    80          BSBW    GETFILE                         ; Get next file
              29 50     E9    004B    81          BLBC    R0,40$                          ; Error if LBC
                 57     95    004E    82          TSTB    R7                              ; Is file spec null (0 bytes)?
                 1B     12    0050    83          BNEQ    30$                             ; No if NEQ
                 5B     D5    0052    84          TSTL    R11                             ; Any files yet?
                 03     12    0054    85          BNEQ    20$                             ; Yes if NEQ
              1E 58     E8    0056    86          BLBS    R8,40$                          ; End of list if LBS
                              0059    87  20$:
                              0059    88          ERRMSG  NULLFS                          ; Report error
            0A          11    006B    89          BRB     40$
                              006D    90  30$:
                 5B     D6    006D    91          INCL    R11                             ; Increment file number
                 07     10    006F    92          BSB     INPUT_SPEC                      ; Process spec
              03 50     E9    0071    93          BLBC    R0,40$                          ; Error if LBC
              CC 58     E9    0074    94          BLBC    R8,10$                          ; More files if LBC
                              0077    95  40$:
                    05        0077    96          RSB
```

```
SUM
Sym

$$
$$.
$$.
..A
..F
..M
..T
.LE
BUF
CLO
CLO
CMD
CMD
CMD
CMD
CRE
DEF
DEF
ED$
ED$
ED$
ED$
ED$
ED$
ED$
ED$
ED$
ED$
FAB
FAB
FAB
FAB
FAB
FAB
FAB
FAB
FAB
FAB
FIL
FIL
GET
GET
GET
GET
GET
INP
INP
INP
INP
LIB
LIB
LIB
LOC
MER
MER
MER
MER
```

```
                                    0078          .SBTTL  INPUT_SPEC
                                    0078  ;
                                    0078  ;
                                    0078  ;
                                    0078  ; Inputs:
                                    0078          R6 = Address of file specification
                                    0078          R7 = Length of file specification
                                    0078  ;
                                    0078  ; Outputs:
                                    0078          R0 = Success/error status
                                    0078  ; Subroutine to process input file spec
                                    0078  ;
                                    0078  ;
                                    0078  INPUT_SPEC:
               0000'CF    DF        0078          PUSHAL  W^VIRT_ADDR                  ; Get slp file node
               0000'CF    DF        007C          PUSHAL  W^SLP_SIZE
       00000000'GF   02   FB        0080          CALLS   #2,G^LIB$GET_VM
                  0B 50   E8        0087          BLBS    R0,10$                       ; OK if LBS
                     50   DD        008A          PUSHL   R0                           ; Signal error
       00000000'GF   01   FB        008C          CALLS   #1,G^LIB$SIGNAL
                     6B   11        0093          BRB     20$
                                    0095  10$:
   00   0000'CF      00   2C        0095          MOVC5   #0,W^0,#0,L^SLP_SIZE, - ; Clear new memory
0000'DF     00000000'EF             009B
                                    00A3                  @W^VIRT_ADDR
        52   0000'CF      D0        00A3          MOVL    W^VIRT_ADDR,R2               ; Set node pointer
           0D A2   5B     90        00A8          MOVB    R11,SLP$B_FILENO(R2)         ; Insert file priority number
       14 A2    18 A2     DE        00AC          MOVAL   SLP$T_AUDST(R2), -           ; Initialise audit string descriptor
                                    00B1                  SLP$Q_AUDDS+4(R2)
       2C A2    30 A2     DE        00B1          MOVAL   SLP$T_AUCST(R2), -           ; Initialise audit string descriptor
                                    00B6                  SLP$Q_AUCDS+4(R2)
                     04   BB        00B6          PUSHR   #^M<R2>                      ; Initialise with default string
     28 A2     0000'CF    B0        00B8          MOVW    W^DEF_AUDIT,SLP$Q_AUCDS(R2)
30 A2   0004'DF   0000'CF  28       00BE          MOVC3   W^DEF_AUDIT,@W^DEF_AUDIT+4, -
                                    00C7                  SLP$T_AUCST(R2)
                     04   BA        00C7          POPR    #^M<R2>
                  53 52   D0        00C9          MOVL    R2,R3                        ; and NAM block pointer
     53    00000048 8F    C0        00CC          ADDL    #SLP$T_NAM,R3
        54   0000'CF      DE        00D3          MOVAL   W^INPUT_FAB,R4
                                    00D8          $FAB_STORE FAB=R4, -                 ; Set up FAB
                                    00D8                  NAM = (R3), -
                                    00D8                  FNA = (R6), FNS = R7
                                    00E4          $NAM_STORE NAM = R3, -
                                    00E4                  BID = #NAM$C_BID, -
                                    00E4                  BLN = #NAM$C_BLN
                     13   10        00EC          BSB     PARSE_SPEC                   ; Parse file spec
                  0F 50   E9        00EE          BLBC    R0,20$                       ; Error if LBC
          0004'DF    62   0E        00F1          INSQUE  (R2),@W^FILE_NODES+4         ; Insert new file node
       30 A4    0C A3     D0        00F6          MOVL    NAM$L_ESA(R3),FAB$L_DNA(R4) ; Reset defaults
       35 A4    0B A3     90        00FB          MOVB    NAM$B_ESL(R3),FAB$B_DNS(R4)
                                    0100  20$:
                          05        0100          RSB
```

```
                      0101          .SBTTL  PARSE_SPEC
                      0101  ;
                      0101  ;
                      0101  ; Subroutine to parse file-spec string and put expanded string
                      0101  ; into dynamic memory buffer
                      0101  ;
                      0101  ; Inputs:
                      0101  ;       R3 = NAM block address
                      0101  ;       R4 = FAB block address
                      0101  ;
                      0101  ; Outputs:
                      0101  ;       R0 = Success/error status
                      0101  ;
                      0101  ;
                      0101 PARSE_SPEC:
              14  BB  0101          PUSHR   #^M<R2,R4>
              47  10  0103          BSB     GET_FS_NODE             ; Get file-spec node
           41 50  E9  0105          BLBC    R0,20$                  ; Error if LBC
                      0108          $NAM_STORE NAM = R3, -
                      0108                  ESA = @W^VIRT_ADDR, ESS = #255
                      0113          $PARSE  FAB = R4                ; Parse file name string
           24 50  E8  011C          BLBS    R0,10$                  ; OK if LBS
           0C A4  DD  011F          PUSHL   FAB$L_STV(R4)           ; Signal error
              50  DD  0122          PUSHL   R0
                      0124          ERRMSG  PRSERR,<R6,R7>
 00000000'GF  02  FB  013A          CALLS   #2,G^LIB$SIGNAL
              06  11  0141          BRB     20$
                      0143 10$:
     52    0B A3  9A  0143          MOVZBL  NAM$B_ESL(R3),R2        ; Get expanded string size
              1F  10  0147          BSB     RETURN_FS_NODE          ; Return unused part of node
                      0149 20$:
              14  BA  0149          POPR    #^M<R2,R4>
              05      014B          RSB
```

```
                                014C          .SBTTL  GET_FS_NODE, RETURN_FS_NODE
                                014C  ;
                                014C  ; Subroutines to get and return file specification node
                                014C  ;
                                014C  ; Get node
                                014C  ;
                                014C  ; Inputs:
                                014C  ;       None
                                014C  ;
                                014C  ; Outputs:
                                014C  ;       R0 = Success/error status
                                014C  ;       VIRT_ADDR = Address of block
                                014C  ;       FILE_SIZE = Size of block
                                014C  ;
                                014C          .ENABL  LSB
                                014C  ;
                                014C  GET_FS_NODE:
  0000'CF   00000100 8F   D0    014C          MOVL    #256,W^FILE_SIZE        ; Set size of expanded string buffer
               0000'CF    DF    0155          PUSHAL  W^VIRT_ADDR             ; Push parameters
               0000'CF    DF    0159          PUSHAL  W^FILE_SIZE
       00000000'GF   02   FB    015D          CALLS   #2,G^LIB$GET_VM
                 25  50   E9    0164          BLBC    R0,10$                  ; Error if LBC
                          05    0167          RSB
                                0168  ;
                                0168  ;
                                0168  ; Return node
                                0168  ;
                                0168  ; Inputs:
                                0168  ;       R2 = Number of bytes in node used
                                0168  ;       VIRT_ADDR = Address of node
                                0168  ;       FILE_SIZE = Size of node
                                0168  ;
                                0168  ; Outputs:
                                0168  ;       R0 = Success/error status
                                0168  ;       VIRT_ADDR = Address of memory returned
                                0168  ;       FILE_SIZE = Size of mempory returned
                                0168  ;
                                0168  ;
                                0168  RETURN_FS_NODE:
                 52  07   C0    0168          ADDL2   #7,R2                   ; Round up to quadword
                 52  07   CA    016B          BICL2   #7,R2
          0000'CF    52   C2    016E          SUBL2   R2,W^FILE_SIZE          ; Compute number of bytes to return
                     20   13    0173          BEQL    20$                     ; None if EQL
          0000'CF    52   C0    0175          ADDL2   R2,W^VIRT_ADDR          ; Address of bytes to return
               0000'CF    DF    017A          PUSHAL  W^VIRT_ADDR             ; Push parameters
               0000'CF    DF    017E          PUSHAL  W^FILE_SIZE
       00000000'GF   02   FB    0182          CALLS   #2,G^LIB$FREE_VM
                 09  50   E8    0189          BLBS    R0,20$                  ; OK if LBS
                                018C  10$:
                     50   DD    018C          PUSHL   R0                      ; Signal error
       00000000'GF   01   FB    018E          CALLS   #1,G^LIB$SIGNAL
                                0195  20$:
                          05    0195          RSB
                                0196  ;
                                0196          .DSABL  LSB
```

```
                              0196   240            .SBTTL  OUTPUT_FILE
                              0196   241  ;
                              0196   242  ;
                              0196   243  ; Subroutine to get output file
                              0196   244  ;
                              0196   245  OUTPUT_FILE:
         56    0000'CF    DE  0196   246            MOVAL   W^INPUT_BUF,R6             ; Get address to put file name string
                 0058     30  019B   247            BSBW    GETFILE                    ; Get next file
                   57     D5  019E   248            TSTL    R7                         ; Is file spec null (0 bytes)
                   35     13  01A0   249            BEQL    20$                        ; Yes if EQL
                1E 58     E9  01A2   250            BLBC    R8,10$                     ; Error if not last file
 0000'CF       0000'8F    A8  01A5   251            BISW    #MERM_OUTPUT,W^MERGE_FLAGS ; Flag output file specified
      53       0000'CF    DE  01AC   252            MOVAL   W^OUTPUT_NAM,R3            ; Set NAM and FAB addresses
      54       0000'CF    DE  01B1   253            MOVAL   W^OUTPUT_FAB,R4
                              01B6   254            $FAB_STORE FAB = R4, FNA = (R6), FNS = R7
                 FF40     30  01BE   255            BSBW    PARSE_SPEC
                   32     11  01C1   256            BRB     40$
                              01C3   257  10$:
                              01C3   258            ERRMSG  ONEOUT
                   1E     11  01D5   259            BRB     40$
                              01D7   260  20$:
                09 58     E9  01D7   261            BLBC    R8,30$                     ; Not at end of line if LBC
 0000'CF       0000'8F    AA  01DA   262            BICW    #MERM_OUTPUT,W^MERGE_FLAGS ; Flag no output file
                   12     11  01E1   263            BRB     40$
                              01E3   264  30$:
                              01E3   265            ERRMSG  NULLFS                     ; Report error
                              01F5   266  40$:
                   05         01F5   267            RSB
```

```
                              01F6        .SBTTL  GETFILE
                              01F6  ;
                              01F6  ;
                              01F6  ; Subroutine to get next file spec from command line
                              01F6  ;
                              01F6  ; Inputs:
                              01F6  ;         R6  = Address to put file spec string
                              01F6  ;
                              01F6  ; Outputs:
                              01F6  ;         R0  = Success/error status
                              01F6  ;         R6  = Address of file-spec
                              01F6  ;         R7  = Size in bytes of file-spec
                              01F6  ;         R8  = Continue/terminate flag
                              01F6  ;
                              01F6 GETFILE:
            0040 8F     BB    01F6          PUSHR   #^M<R6>
                 57     D4    01FA          CLRL    R7                       ; file-spec sting
                 53     D4    01FC          CLRL    R3                       ; Initialise [..] flag
                              01FE 10$:
                 7B     10    01FE          BSB     GETCHAR                  ; Get next character
              6C 50     E9    0200          BLBC    R0,150$                  ; Error if LBC
                 60     13    0203          BEQL    120$                     ; End of line if EQL
    0274'CF   02 55     3A    0205          LOCC    R5,#2,W^LOCCHAR          ; Space or tab?
                 F1     12    020B          BNEQ    10$                      ; Yes if NEQ
                 07     11    020D          BRB     30$
                              020F 20$:
                 6A     10    020F          BSB     GETCHAR                  ; Get next character
              5B 50     E9    0211          BLBC    R0,150$                  ; Error if LBC
                 4F     13    0214          BEQL    120$                     ; End of line if EQL
                              0216 30$:
    0274'CF   07 55     3A    0216          LOCC    R5,#7,W^LOCCHAR          ; Special character
          07  00 50     8F    021C          CASEB   R0,#0,#7
                      001D'   0220 40$:     .WORD   80$-40$                  ; Normal character
                      0010'   0222          .WORD   50$-40$                  ; >
                      0014'   0224          .WORD   60$-40$                  ; <
                      0010'   0226          .WORD   50$-40$                  ; ]
                      0014'   0228          .WORD   60$-40$                  ; [
                      0019'   022A          .WORD   70$-40$                  ; ,
                      0024'   022C          .WORD   90$-40$                  ; Space
                      0024'   022E          .WORD   90$-40$                  ; Tab
                              0230 50$:
                 53     D4    0230          CLRL    R3                       ; Clear [..] flag
                 09     11    0232          BRB     80$
                              0234 60$:
              53 01     D0    0234          MOVL    #1,R3                    ; Set [..] flag
                 04     11    0237          BRB     80$
                              0239 70$:
           2D 53 00     E5    0239          BBCC    #0,R3,130$               ; If ',' but in [..] process as normal
                              023D 80$:
              86 55     90    023D          MOVB    R5,(R6)+                 ; Copy byte to file-spec string
                 57     D6    0240          INCL    R7                       ; and increment size
                 CB     11    0242          BRB     20$                      ; Back for next character
                              0244 90$:
                 35     10    0244          BSB     GETCHAR                  ; Get next character
              26 50     E9    0246          BLBC    R0,150$                  ; Error if LBC
                 1A     13    0249          BEQL    120$                     ; End of line if EQL
    0274'CF   03 55     3A    024B          LOCC    R5,#3,W^LOCCHAR          ; Trailing character?
```

```
          03   00   50    8F  0251     326           CASEB   R0,#0,#3
                       0008'  0255     327 100$:     .WORD   110$-100$                ; No
                       0015'  0257     328           .WORD   130$-100$                ; ,
                        FFEF  0259     329           .WORD   90$-100$                 ; Space
                        FFEF  025B     330           .WORD   90$-100$                 ; Tab
                              025D     331 110$:
               0000'CF    D7  025D     332           DECL    W^CMD_INPUT_POS          ; Back-up line pointer
               0000'CF    D6  0261     333           INCL    W^CMD_INPUT_SZE
                              0265     334 120$:
               58   01    D0  0265     335           MOVL    #1,R8                    ; Set for no more input files
                    02    11  0268     336           BRB     140$
                              026A     337 130$:
                    58    D4  026A     338           CLRL    R8                       ; Set for more input files
                              026C     339 140$:
               50   01    D0  026C     340           MOVL    #1,R0
                              026F     341 150$:
               0040 8F    BA  026F     342           POPR    #^M<R6>
                          05  0273     343           RSB
                              0274     344 ;
    3E 3C 5D 5B 2C 20 09      0274     345 LOCCHAR:          .ASCII  <^X9>/ ,[]<>/
```

```
                         027B    347            .SBTTL  GETCHAR
                         027B    348    ;
                         027B    349    ;
                         027B    350    ; Subroutine to get next character from command line
                         027B    351    ;
                         027B    352    ; Inputs:
                         027B    353    ;           None
                         027B    354    ;
                         027B    355    ; Outputs:
                         027B    356    ;           R0  = Success/error status
                         027B    357    ;           R5  = character
                         027B    358    ;           'Z' = 0 if end of line
                         027B    359    ;           'Z' = 1 if valid character in R5
                         027B    360    ;
                         027B    361    ;
                         027B    362    GETCHAR:
        0300 8F     BB   027B    363            PUSHR   #^M<R8,R9>
          50  01    D0   027F    364            MOVL    #1,R0                           ; Assume success
    59  0000'CF     D0   0282    365            MOVL    W^CMD_INPUT_SZE,R9              ; Set command size
    58  0000'CF     D0   0287    366            MOVL    W^CMD_INPUT_POS,R8              ; Set command input position
              29    12   028C    367            BNEQ    30$                             ; Have a command line if NEQ
                         028E    368    10$:
                         028E    369            $GET    RAB = CMD_INPUT_RAB             ; Prompt for and get next command line
          0F  50    E8   029B    370            BLBS    R0,20$                          ; OK if LBS
         000C'CF    DD   029E    371            PUSHL   W^CMD_INPUT_RAB+RAB$L_STV ; Signal error
              50    DD   02A2    372            PUSHL   R0
 00000000'GF  02    FB   02A4    373            CALLS   #2,G^LIB$SIGNAL
              50    11   02AB    374            BRB     70$
                         02AD    375    20$:
    58  0028'CF     D0   02AD    376            MOVL    W^CMD_INPUT_RAB+RAB$L_RBF,R8 ; Reset command line position
    59  0022'CF     3C   02B2    377            MOVZWL  W^CMD_INPUT_RAB+RAB$W_RSZ,R9 ; and size
                         02B7    378    30$:
              59    D5   02B7    379            TSTL    R9                              ; Any characters in line?
              1E    13   02B9    380            BEQL    40$                             ; No if EQL
          55  88    90   02BB    381            MOVB    (R8)+,R5                        ; Get character
              59    D7   02BE    382            DECL    R9                              ; Decrement character count
          2D  55    91   02C0    383            CMPB    R5,#^A/-/                       ; Continuation character?
              2E    12   02C3    384            BNEQ    60$                             ; No if not equal
              59    D5   02C5    385            TSTL    R9                              ; Last character on line?
              16    12   02C7    386            BNEQ    50$                             ; No if NEQ
0030'CF  0001'CF    DE   02C9    387            MOVAL   W^PROMPT CONT+1, -              ; Set continuation prompt
                         02D0    388                    W^CMD_INPUT RAB+RAB$L_PBF
0034'CF  0000'CF    90   02D0    389            MOVB    W^PROMPT CONT, -
                         02D7    390                    W^CMD_INPUT_RAB+RAB$B_PSZ
              B5    11   02D7    391            BRB     10$
                         02D9    392    40$:
              55    D4   02D9    393            CLRL    R5                              ; Clear character
              58    D4   02DB    394            CLRL    R8                              ; Clear valid command line flag
              14    11   02DD    395            BRB     60$
                         02DF    396    50$:
                         02DF    397            ERRMSG  INVPMD                          ; Issue error message
              0A    11   02F1    398            BRB     70$
                         02F3    399    60$:
 0000'CF      58    D0   02F3    400            MOVL    R8,W^CMD_INPUT_POS              ; Save command position
 0000'CF      59    D0   02F8    401            MOVL    R9,W^CMD_INPUT_SZE              ; and size
                         02FD    402    70$:
              55    D5   02FD    403            TSTL    R5                              ; Set condition codes
```

```
                 0300 8F   BA  02FF    404          POPR     #^M<R8,R9>
                           05  0303    405          RSB
```

```
                              0304   407            .SBTTL  OPEN_FILES
                              0304   408    ;
                              0304   409    ;
                              0304   410    ; Procedure to open slipr input and output files
                              0304   411    ;
                              0304   412    ; Inputs:
                              0304   413    ;       R11 = number of input files
                              0304   414    ;
                              0304   415    ; Outputs:
                              0304   416    ;       None
                              0304   417    ;
                              0304   418    ;
                              0304   419    OPEN_FILES::
                        0000  0304   420            .WORD   0
          5A   0000'CF   DE   0306   421            MOVAL   W^FILE_NODES,R10            ; Initialise file nodes pointer
                              030B   422    10$:
               5A   6A   D0   030B   423            MOVL    (R10),R10                   ; Get next node
       00000000'8F  5A   D1   030E   424            CMPL    R10,#FILE_NODES             ; At end of list?
                    07   13   0315   425            BEQL    20$                         ; Yes if EQL
                    1B   10   0317   426            BSB     OPEN_INPUT                  ; Open input file
               EF   50   E8   0319   427            BLBS    R0,10$                      ; OK if LBC
                    15   11   031C   428            BRB     30$
                              031E   429    20$:
  0004'CF   01000000 8F  C8   031E   430            BISL    #FAB$M_NAM,W^INPUT_FAB+FAB$L_FOP
    0000'CF    0000'8F   B3   0327   431            BITW    #MERM_OUTPUT,W^MERGE_FLAGS ; Was output file specified?
                    03   13   032E   432            BEQL    30$                         ; No if EQL
                  0099   30   0330   433            BSBW    CREATE_OUTPUT               ; Create output file
                              0333   434    30$:
                         04   0333   435            RET
```

```
                            0334             .SBTTL  OPEN_INPUT
                            0334     ;
                            0334     ; Subroutine to open input file
                            0334     ;
                            0334     ; Inputs:
                            0334     ;       R10 = File node address
                            0334     ;
                            0334     ; Outputs:
                            0334     ;       R0 = Success/error code
                            0334     ;
                            0334     ;
                            0334     OPEN_INPUT:
           53    5A   D0    0334             MOVL     R10,R3                     ; Set NAM block address
   53   00000048 8F   C0    0337             ADDL     #SLP$T_NAM,R3
     54   0000'CF     DE    033E             MOVAL    W^INPUT_FAB,R4             ; and FAB address
             FE06     30    0343             BSBW     GET_FS_NODE                ; Get node for resultant file spec
           7E 50      E9    0346             BLBC     R0,30$                     ; Error if LBC
                            0349             $FAB_STORE FAB = R4, NAM = (R3), -
                            0349                      FNA = @NAM$L_ESA(R3), FNS = NAM$B_ESL(R3)
                            0357             $NAM_STORE NAM = R3, ESS = #0, -
                            0357                      RSA = @VIRT_ADDR, RSS = #255
                            0367             $OPEN    FAB = R4                   ; Open input file
           29 50      E9    0370             BLBC     R0,20$                     ; Error if LBC
                            0373             $CLOSE   FAB = R4                   ; Close file to release FAB
           1D 50      E9    037C             BLBC     R0,20$                     ; Error if LBC
     52    03 A3      9A    037F             MOVZBL   NAM$B_RSL(R3),R2           ; Get number of bytes used
             FDE2     30    0383             BSBW     RETURN_FS_NODE             ; and return rest of node
           3E 50      E9    0386             BLBC     R0,30$                     ; Error if LBC
               52     D4    0389             CLRL     R2                         ; Return Expanded fs node
 0000'CF   2C A4      D0    038B             MOVL     FAB$L_FNA(R4),W^VIRT_ADDR
 0000'CF   34 A4      9A    0391             MOVZBL   FAB$B_FNS(R4),W^FILE_SIZE
             FDCE     30    0397             BSBW     RETURN_FS_NODE
               2B     11    039A             BRB      30$
                            039C     20$:
     56    2C A4      D0    039C             MOVL     FAB$L_FNA(R4),R6           ; Get file spec
     57    34 A4      9A    03A0             MOVZBL   FAB$B_FNS(R4),R7
                            03A4             ERRMSG   OPENER,<R6,R7>
           0C A4      DD    03BA             PUSHL    FAB$L_STV(R4)              ; Signal error
           08 A4      DD    03BD             PUSHL    FAB$L_STS(R4)
00000000'GF    02     FB    03C0             CALLS    #2,G^LIB$SIGNAL
                            03C7     30$:
        000C CA       94    03C7             CLRB     W^SLP$B_FLAGS(R10)         ; Initialise flags
                      05    03CB             RSB
```

```
                          03CC          .SBTTL  CREATE_OUTPUT
                          03CC  ;
                          03CC  ; Subroutine to create output file
                          03CC  ;
                          03CC  ; Inputs:
                          03CC          None
                          03CC  ;
                          03CC  ; Outputs:
                          03CC          R0 = Success/error status
                          03CC  ;
                          03CC  ;
                          03CC  CREATE_OUTPUT:
       53  0000'CF    DE  03CC          MOVAL   W^OUTPUT_NAM,R3             ; Set NAM and
       54  0000'CF    DE  03D1          MOVAL   W^OUTPUT_FAB,R4             ; FAB pointers
                          03D6          $FAB_STORE FAB = R4, -
                          03D6                  FNA = @NAM$L_ESA(R3), FNS = NAM$B_ESL(R3)
             FD69     30  03E0          BSBW    GET_FS_NODE                 ; Get file_spec node
             7C 50    E9  03E3          BLBC    R0,40$                      ; Error if LBC
                          03E6          $NAM_STORE NAM = R3, ESS = #0, -
                          03E6                  RSA = @VIRT_ADDR, RSS = #255
                          03F6          $CREATE FAB = R4                    ; Open output file
             05 50    E8  03FF          BLBS    R0,10$                      ; OK if LBS
             0C A4    DD  0402          PUSHL   FAB$L_STV(R4)               ; Signal error
                14    11  0405          BRB     20$
                          0407  10$:
                          0407          $CONNECT RAB = OUTPUT_RAB           ; Connect RAB to FAB
             30 50    E8  0414          BLBS    R0,30$                      ; OK if LBS
           000C'CF    DD  0417          PUSHL   W^OUTPUT_RAB+RAB$L_STV      ; Signal error
                          041B  20$:
                50    DD  041B          PUSHL   R0
       56     2C A4   D0  041D          MOVL    FAB$L_FNA(R4),R6            ; Get file spec
       57     34 A4   9A  0421          MOVZBL  FAB$B_FNS(R4),R7
                          0425          ERRMSG  CREATE,<R6,R7>
          50     6E   D0  043B          MOVL    (SP),R0                     ; Reset R0
  00000000'GF    02   FB  043E          CALLS   #2,G^LIB$SIGNAL
                1B    11  0445          BRB     40$
                          0447  30$:
       52     03 A3   9A  0447          MOVZBL  NAM$B_RSL(R3),R2            ; Get number of bytes used
             FD1A     30  044B          BSBW    RETURN_FS_NODE              ; and return rest of node
             11 50    E9  044E          BLBC    R0,40$                      ; Error of LBC
                52    D4  0451          CLRL    R2                          ; Return expanded fs node
  0000'CF     2C A4   D0  0453          MOVL    FAB$L_FNA(R4),W^VIRT_ADDR
  0000'CF     34 A4   9A  0459          MOVZBL  FAB$B_FNS(R4),W^FILE_SIZE
             FD06     30  045F          BSBW    RETURN_FS_NODE
                          0462  40$:
                      05  0462          RSB
```

```
                       0463          .SBTTL  CLOSE_FILES
                       0463  ;
                       0463  ;
                       0463  ; Procedure to close files
                       0463  ;
                       0463  ; Inputs:
                       0463  ;       File list
                       0463  ;
                       0463  ; Outputs:
                       0463  ;       None
                       0463  ;
                       0463  ;
                       0463  CLOSE_FILES::
                 0000  0463          .WORD   0
                       0465
52      0000'CF    DE  0465          MOVAL   W^INPUT_FAB,R2
             13    10  046A          BSB     CLOSE
52      0000'CF    DE  046C          MOVAL   W^OUTPUT_FAB,R2
             0C    10  0471          BSB     CLOSE
52      0000'CF    DE  0473          MOVAL   W^RANDOM_FAB,R2
             05    10  0478          BSB     CLOSE
        0000'CF    D4  047A          CLRL    W^RANDOM_FILE
                   04  047E          RET
                       047F  ;
                       047F  ;
                       047F  ; Subroutine to close file
                       047F  ;
                       047F  ; Inputs:
                       047F  ;       R2 = FAB address
                       047F  ;
                       047F  ; Outputs:
                       047F  ;       None
                       047F  ;
                       047F  CLOSE:
          02 A2    B5  047F          TSTW    FAB$W_IFI(R2)           ; Is file open?
             09    13  0482          BEQL    10$                     ; No if EQL
                       0484          $CLOSE  FAB = R2                ; Yes it's open so close it
                       048D  10$:
                   05  048D          RSB
                       048E
                       048E  ;
                       048E  ;
                       048E          .END
```

```
$$              = 00000002              MER$_OPENER       *********   X   02
$$.TMP1         = 00000001              MER$_PRSERR       *********   X   02
$$.TMP2         = 00000052              MERGE_FLAGS       *********   X   02
..AFLG          = 00000000              MERM_OUTPUT       *********   X   02
..FLG           = 00000002              NAM$B_BID       = 00000000
..MOD           = 00000000              NAM$B_BLN       = 00000001
..TYP           = 00000053              NAM$B_ESL       = 0000000B
.LEN            = 00000001              NAM$B_ESS       = 0000000A
BUF_SIZE        = 00000200              NAM$B_RSL       = 00000003
CLOSE             0000047F R      02    NAM$B_RSS       = 00000002
CLOSE_FILES       00000463 RG     02    NAM$C_BID       = 00000002
CMD_INPUT_POS     *********   X   02    NAM$C_BLN       = 00000060
CMD_INPUT_RAB     *********   X   02    NAM$K_BLN       = 00000060
CMD_INPUT_SZE     *********   X   02    NAM$L_ESA       = 0000000C
CMD_SIZE        = 00000084              NAM$L_RSA       = 00000004
CREATE_OUTPUT     000003CC R      02    OPEN_FILES        00000304 RG     02
DEF_AUDIT         *********   X   02    OPEN_INPUT        00000334 R      02
DEF_NAME          *********   X   02    OUTPUT_FAB        *********   X   02
ED$B_FILENO     = 00000019              OUTPUT_FILE       00000196 R      02
ED$B_FLAGS      = 00000018              OUTPUT_NAM        *********   X   02
ED$K_BLN        = 0000001A              OUTPUT_RAB        *********   X   02
ED$L_BWD        = 00000004              PARSE_SPEC        00000101 R      02
ED$L_FILE       = 00000014              PROMPT_CONT       *********   X   02
ED$L_FWD        = 00000000              PROMPT_INPUT      *********   X   02
ED$W_LINES      = 0000000C              PROMPT_OUTPUT     *********   X   02
ED$W_LOC1       = 00000008              RAB$B_PSZ       = 00000034
ED$W_LOC2       = 0000000A              RAB$L_PBF       = 00000030
ED$W_RFA        = 0000000E              RAB$L_RBF       = 00000028
FAB$B_DNS       = 00000035              RAB$L_STV       = 0000000C
FAB$B_FNS       = 00000034              RAB$W_RSZ       = 00000022
FAB$L_DNA       = 00000030              RANDOM_FAB        *********   X   02
FAB$L_FNA       = 0000002C              RANDOM_FILE       *********   X   02
FAB$L_FOP       = 00000004              RETURN_FS_NODE    00000168 R      02
FAB$L_NAM       = 00000028              SLP$B_FILENO    = 0000000D
FAB$L_STS       = 00000008              SLP$B_FLAGS     = 0000000C
FAB$L_STV       = 0000000C              SLP$K_BLN       = 000000A8
FAB$M_NAM       = 01000000              SLP$L_BWD       = 00000004
FAB$W_IFI       = 00000002              SLP$L_FWD       = 00000000
FILE_NODES        *********   X   02    SLP$Q_AUCDS     = 00000028
FILE_SIZE         *********   X   02    SLP$Q_AUDDS     = 00000010
GETCHAR           0000027B R      02    SLP$Q_CMNT      = 00000040
GETFILE           000001F6 R      02    SLP$T_AUCST     = 00000030
GET_FILES         00000000 RG     02    SLP$T_AUDST     = 00000018
GET_FS_NODE       0000014C R      02    SLP$T_NAM       = 00000048
GET_HANDLER       *********   X   02    SLP$W_DOT       = 0000000E
INPUT_BUF         *********   X   02    SLP$W_LOC1      = 00000008
INPUT_FAB         *********   X   02    SLP$W_LOC2      = 0000000A
INPUT_FILES       00000033 R      02    SLP_SIZE          *********   X   02
INPUT_SPEC        00000078 R      02    SYS$CLOSE         *********  GX   02
LIB$FREE_VM       *********   X   02    SYS$CONNECT       *********  GX   02
LIB$GET_VM        *********   X   02    SYS$CREATE        *********  GX   02
LIB$SIGNAL        *********   X   02    SYS$GET           *********  GX   02
LOCCHAR           00000274 R      02    SYS$OPEN          *********  GX   02
MER$_CREATE       *********   X   02    SYS$PARSE         *********  GX   02
MER$_INVPMD       *********   X   02    VIRT_ADDR         *********   X   02
MER$_NULLFS       *********   X   02
MER$_ONEOUT       *********   X   02
```

```
                                                    +-----------------+
                                                    ! Psect synopsis !
                                                    +-----------------+

PSECT name                        Allocation           PSECT No.  Attributes
----------                        ----------           ---------  ----------
.  ABS  .                         00000000  (     0.)  00 (  0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                             00000000  (     0.)  01 (  1.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$CODE                             0000048E  (  1166.)  02 (  2.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD   NOWRT NOVEC BYTE

                                                 +------------------------+
                                                 ! Performance indicators !
                                                 +------------------------+

Phase                       Page faults    CPU Time       Elapsed Time
-----                       -----------    --------       ------------
Initialization                      31    00:00:00.09    00:00:00.54
Command processing                 110    00:00:00.71    00:00:01.61
Pass 1                             304    00:00:11.27    00:00:16.98
Symbol table sort                    0    00:00:00.94    00:00:01.01
Pass 2                             119    00:00:02.46    00:00:03.61
Symbol table output                 14    00:00:00.09    00:00:00.09
Psect synopsis output                2    00:00:00.03    00:00:00.03
Cross-reference output               0    00:00:00.00    00:00:00.00
Assembler run totals               582    00:00:15.60    00:00:23.90

The working set limit was 1200 pages.
57645 bytes (113 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 731 non-local and 48 local symbols.
655 source lines were read in Pass 1, producing 19 object records in Pass 2.
38 pages of virtual memory were used to define 27 macros.

                                                 +--------------------------+
                                                 ! Macro library statistics !
                                                 +--------------------------+

Macro library name                               Macros defined
------------------                               --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                                0
_$255$DUA28:[SYSLIB]STARLET.MLB;2                            23
TOTALS (all libraries)                                       23

969 GETS were required to define 23 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SUMFILES/OBJ=OBJ$:SUMFILES MSRC$:SUMCOM/UPDATE=(ENH$:SUMCOM)+MSRC$:SUMFILES/UPDATE=(ENH$:SUMFILES)+EXECML$/LIB
```

0369 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY